20SK: Exercise #5

Source encoding

Jan Přikryl

November 26, 2018

Introduction

Source encoding: *Remove rendunant information.*

Means: *Reduce message entropy to approach $H(X)$.*

Character-based methods (*entropy encoding*):

- Huffman coding
- Arithmetic coding
- Asymetric Numeral Systems (ANS)

Dictionary-based methods:

- LZ77, LZW
- LZMA, Deflate, etc.

Arithmetic Coding

Nearly optimal Entropy encoding

Huffman code optimal for $p_i = 1/2^k$.

Encode the whole message into a binary fraction:

1. resulting number $n \in [0, 1)$,
2. represented using *arbitrary precision interval arithmetic*.

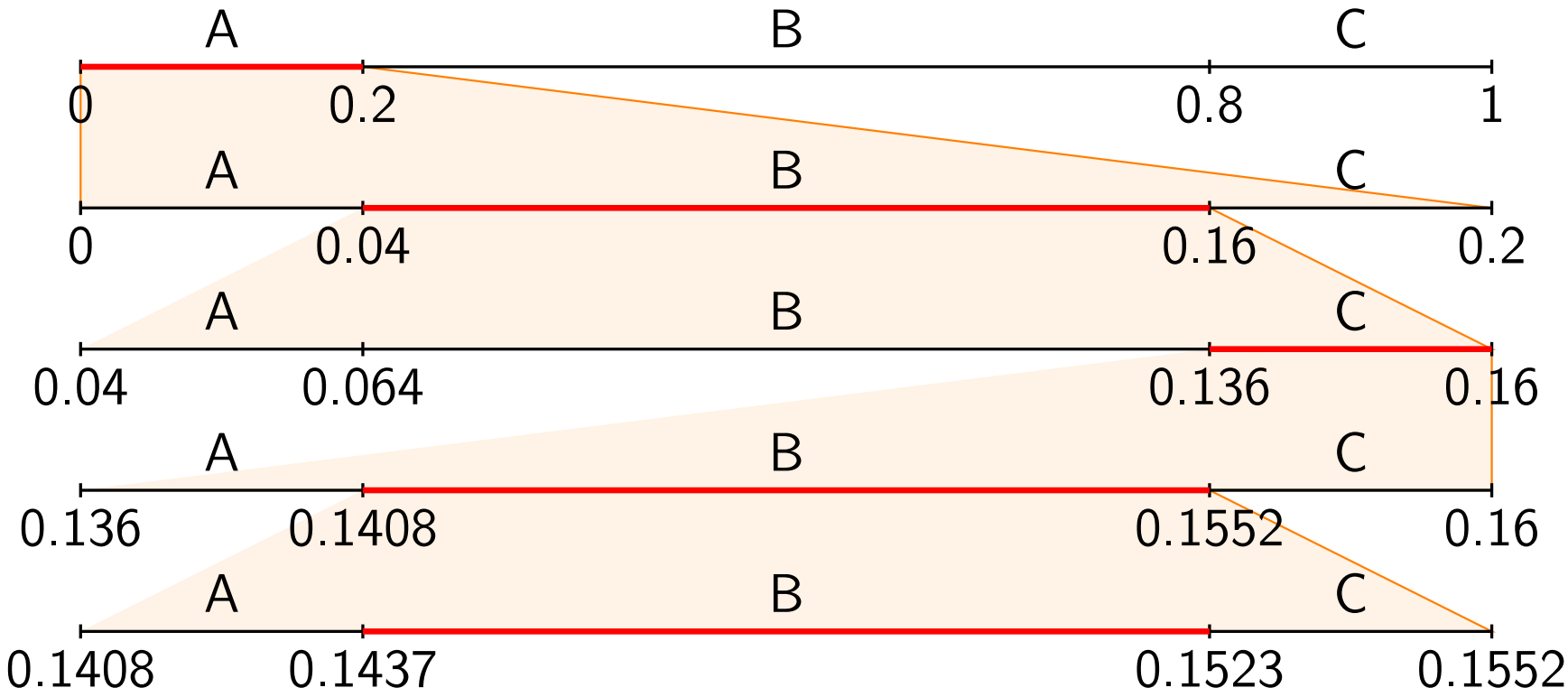
Principle:

1. stochastic model of message data using symbol probabilities, $\sum_i p_i = 1$,
2. *recursive subdivision* of existing sub-intervals.

Arithmetic Coding

Encoding process

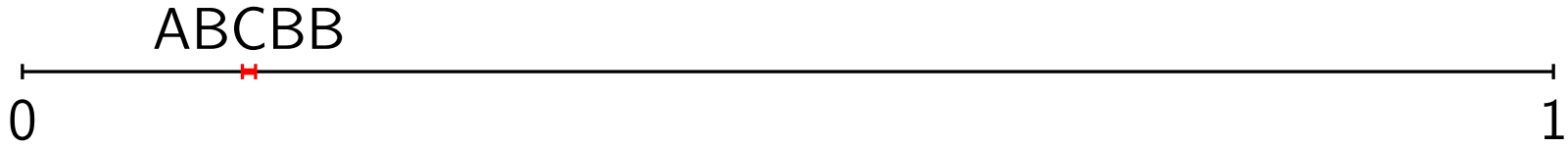
Encode sequence "ABCBB" with $p(A) = 0.2$, $p(B) = 0.6$, $p(C) = 0.2$:



Arithmetic Coding

Resulting message

Encoded sequence "ABCBB" corresponds to $n \in (0.1437, 0.1523)$ in binary form:



Such n is for example $(0.0010011)_b$, i.e.

$$n = \frac{1}{8} + \frac{1}{64} + \frac{1}{128} = 0.1484.$$

Problem 1

1. Create a vector of cumulative probabilities intervals for alphabet $\{A, B, C\}$ with probabilities $p(A) = 0.2$, $p(B) = 0.6$, $p(C) = 0.2$ such that it can be used for arithmetic coding
   - 1.1 write down the elements of the vector for the given alphabet; what number is the first element of the vector, what number will be the last one?
   - 1.2 cumulative probabilities can be computed using `cumsum`, is it enough?
   - 1.3 how will you get the probability interval limits for character A?
   - 1.4 how will you get the probability interval limits for an arbitrary character?

Problem 2

1. Write a function `[p0,p1]=aenc_int(text,probs)` that returns interval $[p_{0,}p_1] \in [0,1]$ for a character sequence in vector `text` with symbol probabilities in `probs`.
   - 1.1 compute the cumulative probabilities including the leading zero
   - 1.2 how will you recompute the cumulative probabilities after processing an input character from `text`?
   - 1.3 how will you get the probability interval limits for an arbitrary character?

Problem 3

1. Write a function `binstr=aenc_bin(p0,p1)` that returns a binary string representing a binary fraction $b \in [p_0, p_1]$
   - 1.1 how do you add a single character to a string (character vector) in Matlab?
   - 1.2 loop over binary fractions $1/2$, $1/4$, ... and accumulate them one by one into a candidate value $\hat{b}$
   - 1.3 what action should be taken if the value of $\hat{b}$ is
     - 1.3.1 below the lower bound p_0,
     - 1.3.2 above the upper bound p_1,
     - 1.3.3 within $[p_0, p_1]$?

Problem 4

1. Create function `binstr=aenc(text,probs)` that combines functions `aenc_int` and `aenc_bin` together.